

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-44563

(43) 公開日 平成8年(1996)2月16日

| | | | | |
|---------------------------|---------|---------|-----|--------|
| (51) Int.Cl. ⁶ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
| G 0 6 F 9/38 | 3 7 0 B | | | |
| | 3 1 0 A | | | |
| 12/08 | H | 7623-5B | | |

審査請求 未請求 請求項の数 1 O L (全 11 頁)

(21) 出願番号 特願平6-178129

(22) 出願日 平成6年(1994)7月29日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 久保沢 元

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74) 代理人 弁理士 長谷川 文廣 (外2名)

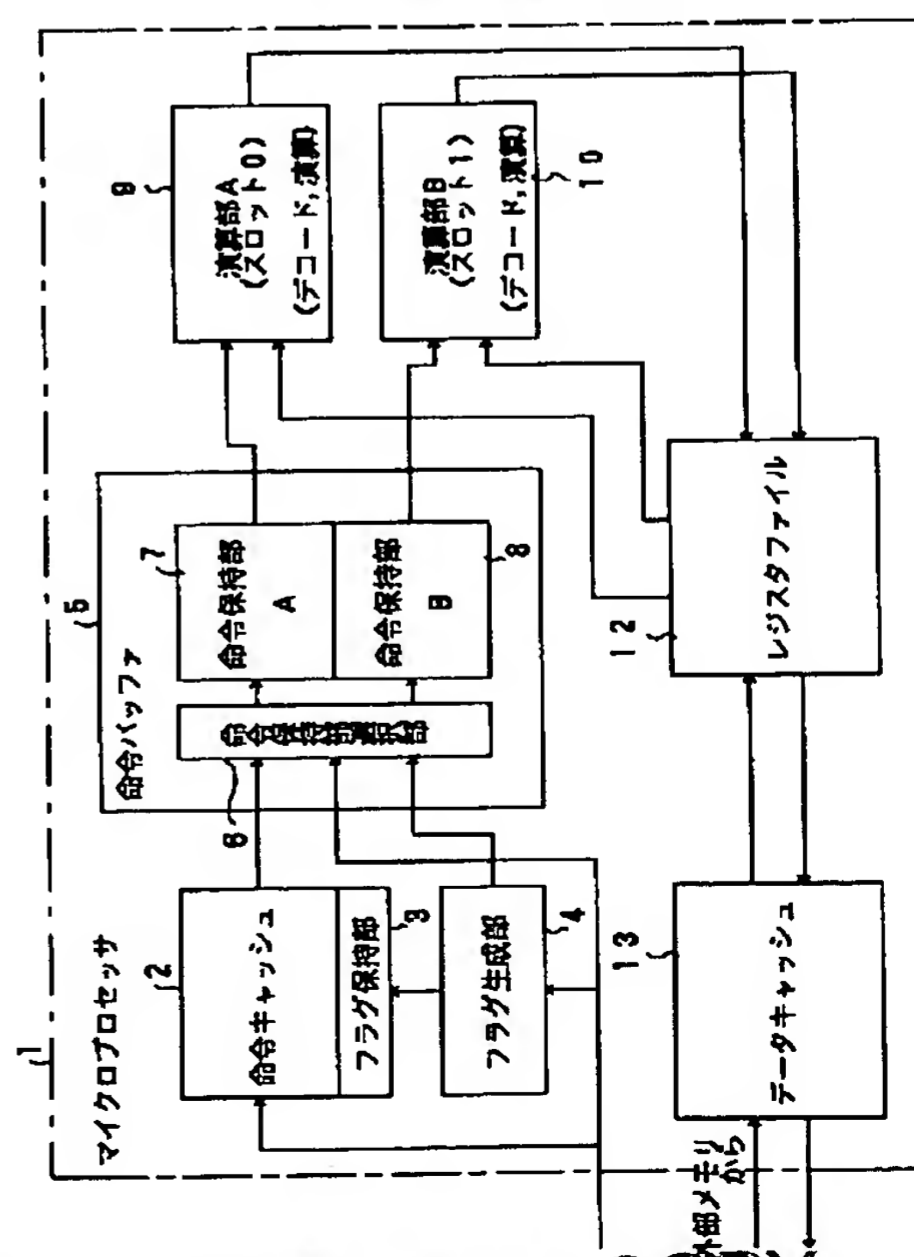
(54) 【発明の名称】 マイクロプロセッサ

(57) 【要約】

【目的】 1クロックサイクルに複数の命令を同時に実行できるマイクロプロセッサに関し、デコードステージでのスロット選択を行わないようにして、高速化を図ることを目的とする。

【構成】 演算処理する命令を保持する命令バッファと命令バッファから命令を入力して演算処理する複数の演算部とを備え、複数の命令実行を同時にできるマイクロプロセッサにおいて、命令は処理する演算部が定められたものであり、命令バッファは、それぞれの演算部に対応して命令保持部を備え、命令が使用する演算部を特定するフラグを生成して命令に付加するフラグ生成部と、フラグに応じて命令バッファの該命令保持部を選択する命令保持選択部とを備え、命令が処理する演算部に対応した命令保持部に該命令を保持する構成を持つ。

本発明の基本構成



BEST AVAILABLE COPY

【特許請求の範囲】

【請求項 1】 演算処理する命令を保持する命令バッファと命令バッファから命令を入力して演算処理する複数の演算部とを備え、複数の命令実行を同時にできるマイクロプロセッサにおいて、命令は処理される演算部が定められたものであり、命令バッファは、それぞれの演算部に対応して命令保持部を備え、命令が使用する演算部を特定するフラグを生成して命令に付加するフラグ生成部と、該フラグに応じて命令バッファの該命令保持部を選択する命令保持部選択部とを備え、命令が処理される演算部に対応した命令保持部に該命令を保持するようにしたことを特徴とするマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、マイクロプロセッサに関する。特に、複数の演算ユニットを有し、1クロックサイクルに複数の命令を同時に実行できるマイクロプロセッサに関する。

【0002】 パイプラインで複数の命令を同時に実行することにより、命令を一つずつ順次実行する場合と比較して高速にプログラム処理することができる。パイプライン処理により複数命令を同時に実行するためには、命令をどの演算器で実行すれば良いかを判定して適切なパイプライン（スロット）に命令を転送する必要がある。

【0003】 本発明は、パイプラインのどのスロットに命令を発行するか判定とスロットへの命令転送を効率良く行うことのできるマイクロプロセッサを提供する。

【0004】

【従来の技術】 図8は従来のパイプライン処理をするマイクロプロセッサの構成を示す。200はプロセッサである。

【0005】 210は入力バッファであって、外部メモリから入力される命令を一時保持するものである。211は命令キャッシュであって、命令を保持するキャッシュである。

【0006】 212はデータキャッシュであって、データを保持するキャッシュである。213は命令バッファであって、命令をパイプライン処理のスロット（スロット0、スロット1）に入力するためのバッファである。

【0007】 214はスロット0であって、パイプライン処理するものである。215はスロット1であって、パイプライン処理するものである。216はレジスタファイルであって、演算器の入力データ（オペランド）、演算器の演算結果等を一時的に保持する複数のレジスタにより構成されるものである。

【0008】 220、221はデコーダであって、命令をデコードするものである。230は演算器1であって、スロット0の演算処理をするものである。231は演算器2であって、スロット0の演算処理をするものである。

【0009】 232は演算器3であって、スロット1の演算処理をするものである。233は演算器4であって、スロット1の演算処理をするものである。図8の従来のマイクロプロセッサの構成の動作を説明する。

【0010】 命令は、命令バッファ213から1サイクルに2命令ずつデコーダ220、221に供給される。命令バッファに命令がある場合は、そこから命令が読み出され、スロット0、スロット1の各デコーダ220、221で解釈される。1命令の語長を4バイトとすると、命令バッファへ213の書き込み、読み出しは8バイト単位で行われる。命令キャッシュ211への書き込み、読み出しの単位も8バイトとする。

【0011】 命令バッファ213に命令がないとき、命令が命令キャッシュ211から読み出されて命令バッファ213に書き込まれる（命令フェッチ）。そして、命令バッファ213の命令はスロット0もしくはスロット1に入力され、デコーダ220、221で解釈される。デコードステージでは命令間のデータ依存性、演算器等の資源の競合がチェックされ、競合がある場合にはそれが解消するまで演算の開始を待たせる。デコードされた命令は、各命令に応じた演算器（演算器1、演算器2、演算器3、演算器4）で演算される。演算を行うための入力オペランドはデコードと並行してレジスタファイル216から読み出される。

【0012】 命令が命令キャッシュ211に存在しない場合には、マイクロプロセッサ200の外部メモリから読み出された命令が命令バッファ213に書き込まれるとともに、命令キャッシュ211にも書き込まれる。命令が演算ステージ（演算器1、演算器2、演算器3、演算器4）に移行した後のフローは命令によって異なる。演算が算術演算、論理演算の場合は、1サイクルで演算は終了し、結果は3サイクル後にレジスタファイル216に書き込まれる。命令がロード命令の場合は、演算実行ステージでメモリにアクセスするアドレスを計算し、次のキャッシュステージでデータキャッシュを読み出す。次のステージで例外処理（例えば、アクセスを禁止されているアドレスが求められてはいないかを検出する等の処理）を行い、次のステージでレジスタの内容を更新する。

【0013】 図8において、演算器1、演算器2、演算器3、演算器4はすべて異なる機能の演算を行うものとする。つまり、同じ種類の演算を2命令同時に実行することはできない。また、スロット0にある命令はスロット1側の演算器を使用することはできない。同様に、スロット1にある命令はスロット0で使用することはできない。これらはスロット間の配線およびマルチプレクサが増加し制御が複雑になることを避けるためである。

【0014】 通常は、命令キャッシュ211からバッファ213に命令を転送してきた時点では、スロット0にはフェッチしてきた8バイトの下位側の命令が、スロッ

ト 1 には上位の命令が入っている。各演算器 (230, 231, 232, 233) に上記のような制限があった場合には、命令がどの演算器を使用するのかを検出して適切なスロットが選ばれる必要がある。使用する演算器と違うスロットに入っている場合には、命令スロットの入れ替えが必要になる。この入れ替えは命令バッファから命令を読み出し、デコードして演算器に渡されるまでの間 (デコードステージ) に行わなければならない。

【0015】

【発明が解決しようとする課題】従来のパイプライン処理するマイクロプロセッサはデコードステージで命令スロットの入れ替えを行うようにしていた。デコードステージは命令のデコード、依存関係のチェック、レジスタファイル読み出し、命令発行が可能であるか等の判定を行うのでプロセッサ全体のクリティカルパス (動作遅延の最も大きいパス) になる可能性の高いステージである。デコードステージで命令がどの演算器を使用するかを判定し、適切なスロットを選択することはマイクロプロセッサの高速動作をさまたげる大きな要因となる。

【0016】本発明は、デコードステージでのスロット選択を行わないようにして、マイクロプロセッサの高速化を図ることを目的とする。

【0017】

【課題を解決するための手段】本発明は、命令キャッシュにチップ外部から命令を格納する際に、命令がどのスロットを使用するかを示すフラグを付け、命令キャッシュにそのフラグとともに書き込む。また、外部メモリから読み出した命令を命令バッファに書き込む場合も同様に、フラグを付け、命令バッファと命令キャッシュに書き込むようにする。そして、命令バッファへの書き込みの際に命令キャッシュのフラグを見て適切なスロットを選択し、命令バッファ内の転送制御はこのスロットフラグによって行うことにより、デコードステージでスロット選択を行わないでもスロットの選択をできるようにした。

【0018】図 1 は本発明の基本構成を示す。図 1 は二つのスロットを備えている場合を例として示す。図 1 において、1 はマイクロプロセッサである。

【0019】2 は命令キャッシュである。3 は命令キャッシュのフラグ保持部であって、命令を演算する演算部 (スロット) を識別するためのフラグを保持するものである。

【0020】4 はフラグ生成部であって、命令の種類に応じてフラグを生成するものである。5 は命令バッファであって、命令キャッシュもしくは外部メモリから読み出した命令を保持するものである。

【0021】6 は命令保持部選択部であって、命令のフラグを識別して、命令保持部 A もしくは命令保持部 B のいずれに命令を保持すべきかを判定するものである。7 は命令保持部 A であって、演算部 A (スロット 0) で処

理される命令を保持するものである。

【0022】8 は命令保持部 B であって、演算部 B (スロット 1) の側で処理される命令を保持するものである。9 は演算部 A (スロット 0) であって、命令保持部 A から入力される命令をデコードし、デコードの結果によりレジスタファイル 12 からオペランドを入力し、演算し、演算結果をレジスタファイル 12 に転送するものである。

【0023】10 は演算部 B (スロット 1) であって、命令保持部 B から入力される命令をデコードし、デコードした結果によりレジスタファイル 12 からオペランドを入力し、演算し、演算結果をレジスタファイル 12 に転送するものである。

【0024】12 はレジスタファイルであって、データキャッシュ 13 から入力されるデータもしくは演算部 A、演算部 B の演算結果を保持するものである。13 はデータキャッシュであって、データを保持するものである。

【0025】

【作用】図 1 の本発明の基本構成の動作を説明する。フラグ生成部 4 は外部メモリから入力される命令をデコードし、命令の種類を判定する。そして、その命令が演算部 A で処理すべきものであるか、あるいは演算部 B で処理すべきものであるかを識別するためのフラグを生成する。例えば、演算部 A で処理する命令に対しては「0」、演算部 B で処理する命令に対しては「1」等である。

【0026】命令バッファ 5 に命令キャッシュ 2 から命令が読み出されて保持される。もしくは、命令キャッシュ 2 に命令が無い場合に外部メモリから入力される命令はフラグ生成部 4 でフラグを付され命令バッファ 5 に入力されるとともに命令キャッシュ 2 にも保持される。その際、命令保持部選択部 6 はフラグを識別し、例えば、フラグが「0」であれば命令バッファ 5 を選択し、命令バッファ A に命令を保持する。あるいは、フラグが

「1」であれば、命令保持部選択部 6 は命令保持部 B を選択し、命令保持部 B に命令を保持する。レジスタファイル 12 はデータキャッシュ 13 からオペランドとなるデータを取り出して保持する。

【0027】演算部 A には命令保持部 A の命令が入力され、入力された命令をデコードする。そして、演算部 A はレジスタファイル 12 から必要なオペランドを取り出し演算し、演算結果をレジスタファイル 12 に保持させる。同様に、演算部 B には命令保持部 B の命令が入力され、入力された命令をデコードする。そして、演算部 B はレジスタファイル 12 から必要とするオペランドを取り出して演算する。そして、演算結果をレジスタファイル 12 に保持する。レジスタファイル 12 に保持された演算結果はデータキャッシュ 13 に転送され、外部メモリに出力される。

【0028】本発明によれば、マイクロプロセッサの処理速度に大きく影響する演算部A、演算部Bのデコードステージでのスロット選択を行わない。そのため、デコードステージで処理が遅延されることがないので、能率的にパイプライン処理を行うことができ、マイクロプロセッサ全体の動作が高速化される。また、命令バッファの命令処理もフラグにより簡単に制御できる。

【0029】

【実施例】図2は本発明の実施例を示す。図2において、51はマイクロプロセッサである。

【0030】52は命令キャッシュである。53はスロットを識別するためのフラグである。フラグ「0」でスロット0、フラグ「1」でスロット1を識別するとする。

【0031】54はプリデコーダであって、外部メモリから命令をデコードしてスロット0もしくはスロット1のいずれで演算すべき命令であるかを識別し、フラグを生成するものである（図1のフラグ生成部に相当する）。

【0032】55は命令バッファである（命令バッファ55の詳細は図4で説明する）。56は命令保持部選択部である。57は命令保持部である。

【0033】59はスロット0であって、デコーダと演算器1、演算器2により構成されるものである。60はスロット1であって、デコーダと演算器3、演算器4により構成されるものである。

【0034】62はレジスタファイルである。63はデータキャッシュである。76は入力バッファ（外部入力レジスタ）であって、外部メモリから入力される命令を一時保持するバッファである。

【0035】スロット0において、70はデコーダであって、スロット0に入力される命令をデコードするものである。

【0036】72は演算器1であって、デコーダ70のデコードした命令の内容に従って、レジスタファイル62からオペランドを取り出し演算処理するものである。73は演算器2であって、デコーダ70のデコードした命令の内容に従って、レジスタファイル62からオペランドを取り出し演算処理するものである。

【0037】スロット1において、71はデコーダであって、スロット1に入力される命令をデコードするものである。

【0038】74は演算器3であって、デコーダ71のデコードした命令の内容に従って、レジスタファイル62からオペランドを取り出し演算処理するものである。75は演算器4であって、デコーダ71のデコードした命令の内容に従って、レジスタファイル62からオペランドを取り出し演算処理するものである。

【0039】80はフラグを付された命令のデータ構造の例である。命令0はスロット0で処理される命令であ

り、スロット0を識別するフラグを持つ。命令1はスロット1で処理される命令であり、スロット1を識別するフラグを持つ。

【0040】命令キャッシュ52の1ラインが32バイト長であるとする8命令分が格納できる。フラグの書き込みは、外部メモリから命令を取ってきた時に行う。外部メモリから読み込んだ命令は、命令キャッシュ52に書き込む前に入力バッファ76（外部入力レジスタ）に一旦書き込みを行う。フラグの発生はこの入力バッファ76を読み出してから命令キャッシュ52に書き込む間にデコーダ54で行われる。命令キャッシュ52に書き込むデータが必要になるタイミングはサイクルの後半である。フラグ発生はこれより速く終わるのでプリデコーダ54によるデコードが動作遅延の原因にはならない（この点については図3で詳述する）。命令キャッシュ52から命令バッファ55への読み出しは命令フェッチステージで行う。命令フェッチステージでは命令キャッシュ52の読み出しが行われ、命令バッファ55に書き込まれる。命令バッファ55において、命令フラグを識別し、命令がスロット0で処理されるべきものであるか、あるいはスロット1で処理されるべきものであるかを命令保持部選択部56で識別し、スロット0で演算すべき命令であれば命令保持部57のスロット0側に保持し、あるいはスロット1で演算すべき命令であればスロット1側に保持する。

【0041】図3は本発明のフラグ生成のタイムチャートである。外部メモリから読み出された命令は一旦、外部入力レジスタ（入力バッファ76）に保持される。

【0042】第1サイクルにおいて、まず、第1サイクルのクロックの立ち上がりで、入力レジスタ76から命令が読み出される。そして、チップ内（マイクロプロセッサ51の内部）を転送され、プリデコーダ54でプリデコードされ、第1サイクルの後半で、チップ内を転送されて命令キャッシュ52の入口の命令キャッシュ入力レジスタ（命令キャッシュ52に命令を入力するために一時命令を保持するレジスタ（図示せず））に書き込まれる。ここで、命令バッファ55にも書き込まれる。

【0043】第2サイクルにおいて、第2サイクルのクロックの立ち上がりで、命令キャッシュの入力レジスタから命令が読み出され、命令キャッシュ52に書き込まれる。

【0044】図4は本発明の命令バッファの実施例である。図4において、52は命令キャッシュである。命令キャッシュには4バイトの命令を8バイト単位に保持される。8バイト構成のLSW（下位）側の命令（命令0）とMSW（上位）側の命令（命令1）にそれぞれスロットを識別するフラグが備えられる。

【0045】55は命令バッファであり、命令の保持部を7個（命令保持部T、P0、P1、F0、F1、D0、D1）備えるものである。命令は、命令の連続性、

フラグを考慮した選択ルールに従って選択的に命令保持部に保持される。選択ルールについては後述する。

【0046】85は命令保持部選択部1であって、命令キャッシュ52もしくは命令保持部Tから取り出された命令のフラグを参照し、選択ルールに従って命令保持部P0もしくはP1を選択するものである。

【0047】86は命令保持部選択部2であって、命令キャッシュ52、命令保持部P0もしくは命令保持部P1から取り出された命令のフラグを参照し、選択ルールに従って命令保持部F0もしくはF1を選択するものである。

【0048】87は命令保持部選択部3であって、命令キャッシュ52、命令保持部F0もしくは命令保持部F1から取り出された命令のフラグを参照し、選択ルールに従って命令保持部D0もしくはD1を選択するものである。

【0049】90はセクタ0であって、P0に保持する命令を選択するものである。91はセクタ1であって、P1に保持する命令を選択するものである。92はセクタ2であって、F0に保持する命令を選択するものである。

【0050】93はセクタ3であって、F1に保持する命令を選択するものである。94はセクタ4であって、D0に保持する命令を選択するものである。95はセクタ5であって、D1に保持する命令を選択するものである。

【0051】Tは命令保持部であって、命令キャッシュ52の命令の上位4バイトの命令を保持するものである。P0は命令保持部であって、セクタ0(90)で選択された命令を保持するものである。

【0052】P1は命令保持部であって、セクタ1(91)で選択された命令を保持するものである。F0は命令保持部であって、セクタ2(92)で選択された命令を保持するものである。

【0053】F1は命令保持部であって、セクタ3(93)で選択された命令を保持するものである。D0は命令保持部であって、セクタ4(94)で選択された命令を保持するものである。

【0054】D1は命令保持部であって、セクタ5(95)で選択された命令を保持するものである。図4の構成において、セクタ0(90)には、命令保持部T、命令キャッシュのLSW(命令0)、MSW(命令1)の内容とフラグが入力され、フラグおよび選択ルールに従って、それらの命令のいずれかを選択する。そして、P0に選択した命令0を保持する。同様に、セクタ1(91)には、命令保持部T、命令キャッシュ52の命令のLSW、MSWの内容とフラグが入力され、フラグおよび選択ルールに従って選択されてP1に保持される。

【0055】また、セクタ2(92)には、命令保持

部P0、命令キャッシュ52のLSW、MSWの内容とそのフラグが入力され、フラグと選択ルールに従って、そのいずれかを選択し、命令保持部F0に保持する。同様に、セクタ3(93)には、命令保持部P1、命令キャッシュ52のLSW、MSWの内容とフラグが入力され、フラグと選択ルールに従って命令が選択されて命令保持部F1に保持される。

【0056】また、セクタ4(94)には、命令保持部F0、命令キャッシュ52のLSW、MSWの内容とフラグが入力され、フラグと選択ルールに従って命令が選択されて、命令保持部D0に保持される。同様に、セクタ5(95)には、命令保持部F1、命令キャッシュ52のLSW、MSWの内容とフラグが入力され、フラグと選択ルールに従って命令が選択されて命令保持部D1に保持される。

【0057】D0の内容はスロット0に転送され、D1の内容はスロット1に転送される。次に図4の構成における命令の転送ルールについて説明する。図4の構成において、各選択部は次のルールに従って、命令を選択し、選択した保持部に転送する。

【0058】(1) キャッシュからの命令の読み出しは2命令単位で行う。

(2) D0、F0、P0にはスロット0の演算器のグループで実行される命令が入る。

【0059】(3) D1、F1、P1にはスロット1の演算器のグループで実行される命令が入る。

(4) 読み出した命令がスロット0で実行する命令とスロット1で実行する命令のペアの場合において、(4)-1 命令バッファが空の時はD0、D1に書き込む。

【0060】(4)-2 D0、D1に命令が存在している時はF0、F1に書き込む。

(4)-3 D0、D1、F0、F1に命令が存在している時はP0、Tに書き込む。

【0061】(5) 読み出した命令が両方ともスロット0で実行する命令の場合、(5)-1 命令バッファが空の時はD0、F0に書き込む。(5)-2 D0、D1に命令が存在している時はF0、P0に書き込む。

【0062】(5)-3 D0、D1、F0、F1に命令が存在している時はP0、Tに書き込む。

(6) 読み出した命令が両方ともスロット1で実行する命令の場合、(6)-1 命令バッファが空の時はD1、F1に書き込む。

【0063】(6)-2 D0、D1に命令が存在している時はF1、P1に書き込む。(6)-3 D0、D1、F0、F1に命令が存在している時はP1、Tに書き込む。

【0064】(7) D0、D1から命令が発行されると、F0、F1、P0、P1、Tに存在する命令はD0、D1の方向に詰めるようにバッファ内を移動する。

(8) D0、D1に同時に入る命令は連続したアドレス

の命令でなければならない。

【0065】(9) F0, F1に同時に入る命令は連続したアドレスの命令でなければならない。

(10) P0, P1に同時に入る命令は連続したアドレスの命令でなければならない。

【0066】図4の命令バッファの動作を動作例に基づいて説明する(図5, 図6, 図7を参照する)。図5は命令キャッシュ52に保持されている命令列の例である。4バイト構成の2命令を1サイクルの処理単位とし、8バイトの上位4バイトのMSWと下位4バイトのLSWをそれぞれ1命令とする。それぞれにスロットを識別するフラグを備えている。フラグ「0」はスロット0で処理される命令であり、フラグ「1」はスロット「1」で処理される命令である(add命令はスロット0で実行され、load命令はスロット1で実行されたものとする。)

命令キャッシュ52から各クロックサイクル(以下サイクルと略称する)毎に2命令もしくは、上位バイト(MSW)が読み出される。同じサイクルでバッファ(命令保持部)に書き込まれた命令は、同時に実行できる。バッファに書き込まれるサイクルが異なる場合は、命令のアドレスが4バイト境界で連続している命令は同時に実行できる(例えば、上記の命令で、load2とadd3, add4とload3等)。そうでないもの同士(例えば、load2とload4等)は同時に実行できない。

【0067】図6は動作例1である。サイクル1で命令キャッシュ52からadd1とload1が読み出され、セクタ4(94)とセクタ5(95)で選択されてD0とD1に書き込まれる。ここで、2命令を実行する命令が発行される(2命令発行)。

【0068】サイクル2で、2命令発行に従って、add1とload1がそれぞれD0, D1からスロット0, スロット1に出力される。そして、新たにload2とadd2が命令キャッシュ52から読み出され、セクタ4(94), セクタ5(95)で選択され、それぞれD0, D1に書き込まれる。ここで2命令実行が発行される。

【0069】サイクル3で、2命令発行に従って、load2とadd2がそれぞれスロット0, スロット1に出力される。そして、新たにadd3とadd4が命令キャッシュ52から読み出される。両方ともスロット0の命令であるので、セクタ4(94), セクタ2(92)で選択されてD0, F0に書き込まれる。ここで1命令の実行が発行される(1命令発行)。

【0070】サイクル4で、1命令発行に従ってadd3がD0からスロット0に出力され、セクタ4(94)で選択されてadd4がF0からD0に移動する。ここで、次の命令load3とload4が命令キャッシュ52から読み出される。load3とadd4は境

界が連続している(プログラムカウンタ値が連続している)ので、同時に実行できる。そのため、load3はセクタ5(95)で選択されてD1に保持され、load4はセクタ3(93)で選択されてF1に保持される。ここで、2命令実行が発行される。

【0071】サイクル5で、2命令発行のためload3とadd4がそれぞれスロット0, スロット1に出力される。そして、セクタ5(95)で選択されてload4がF1からD1に移動し、新たに命令キャッシュ52からload5, add5が読み込まれる。そして、D1のload4とadd5は同時に実行できないので、add5はセクタ2(92)で選択されてF0に書き込まれ、load5はセクタ3(93)で選択されてF1に書き込まれる。ここで1命令実行が発行される。

【0072】サイクル6で、1命令発行のためload4がD1からスロット1に出力され、load5, add5がそれぞれセクタ5(95), セクタ4(94)で選択されてD1, D0に移動する。新たに、命令キャッシュ52から、load6, add6が読み出され、それぞれセクタ3(93), セクタ2(92)で選択されてF1とF0に書き込まれる。ここで、2命令実行が発行され、次のサイクル(図示せず)でload5とadd5が出力される。

【0073】図7は動作例2を示す。以下、セクタの選択処理についての説明は省略する。サイクル1で命令キャッシュからadd1とload1が読み出され、D0, D1に書き込まれる。ここで、0命令発行で、次のサイクルで2命令とも実行されないとする。

【0074】サイクル2でadd1とload1はD0とD1に保持される。そして、次の命令(load2とadd2)が命令キャッシュ52から読み出され、F1とF0に保持される。0命令発行で、サイクル3でも命令が実行されないとする。

【0075】サイクル3でadd3とadd4が命令キャッシュ52から読み出され、上位バイトのadd4はTに保持され、下位バイトのadd3はP0に保持される。サイクル3で2命令発行がなされる。

【0076】サイクル4でload1とadd1がそれぞれD1, D0からスロット1とスロット0に出力される。そして、load2とadd2がそれぞれD1, D0に移動する。add3とadd4はそれぞれF0, P0に移動する。命令キャッシュ52から、新たにload3とload5が読み込まれる。load3はadd4と境界が連続しているためP1に書き込まれ、load4はTに書き込まれる。ここで、2命令発行がなされる。

【0077】サイクル5で、2命令発行に従って、load2とadd2がそれぞれD1, D0からスロット1, スロット0に出力される。そして、add3, a

d 4はそれぞれD 0, P 0に移動する。また, l o a d 3, l o a d 4はそれぞれF 1, P 1に移動する。ここで, 次の命令l o a d 5とa d d 5を読み出す余地がないので, 新たに命令は読み出されない。ここで1命令発行がなされる。

【0078】サイクル6で, 1命令発行に従って, a d d 3がD 0から出力され, a d d 4がD 0に移動する。そして, l o a d 3, l o a d 4がそれぞれD 1, F 1に移動する。命令キャッシュ52から新たに, l o a d 5とa d d 5が読み出され, それぞれP 1, P 0に保持される。ここで2命令発行がなされる。

【0079】サイクル7で, l o a d 3とa d d 4がそれぞれD 0, D 1からスロット1, スロット0に出力される。そして, l o a d 4がD 1に移動し, l o a d 5, a d d 5がそれぞれF 1, F 0に移動する。命令キャッシュ52からl o a d 6, a d d 6が読み出され, それぞれP 1, P 0に保持される。ここで1命令が発行される。

【0080】

【発明の効果】本発明によれば, マイクロプロセッサの処理速度に大きく影響する演算部A, 演算部Bのデコードステージでのスロット選択を行わない。そのため, デコードステージで処理が遅延されることがないので, 能率的にパイプライン処理を行うことができ, マイクロプロセッサ全体の動作が高速化される。また, 命令バッファの命令処理もフラグにより簡単に制御できる。

【図面の簡単な説明】

【図1】本発明の基本構成を示す図である。

【図2】本発明の実施例を示す図である。

【図3】本発明のフラグ生成のタイムチャートを示す図である。

【図4】本発明の命令バッファの実施例を示す図である。

【図5】本発明の命令キャッシュに保持されている命令の例を示す図である。

【図6】本発明の動作例1を示す図である。

【図7】本発明の動作例2を示す図である。

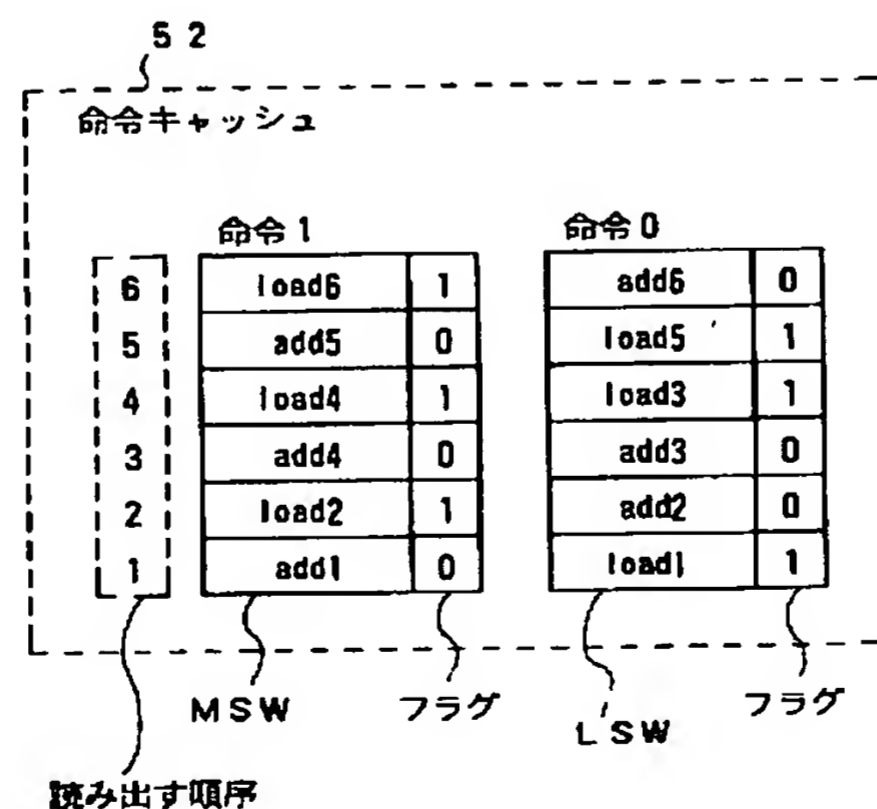
【図8】従来のマイクロプロセッサの構成を示す図である。

【符号の説明】

- 1：マイクロプロセッサ
- 2：命令キャッシュ
- 3：フラグ保持部
- 4：フラグ生成部
- 5：命令バッファ
- 6：命令保持部選択部
- 7：命令保持部A
- 8：命令保持部B
- 9：演算部A
- 10：演算部B
- 12：レジスタファイル
- 13：データキャッシュ

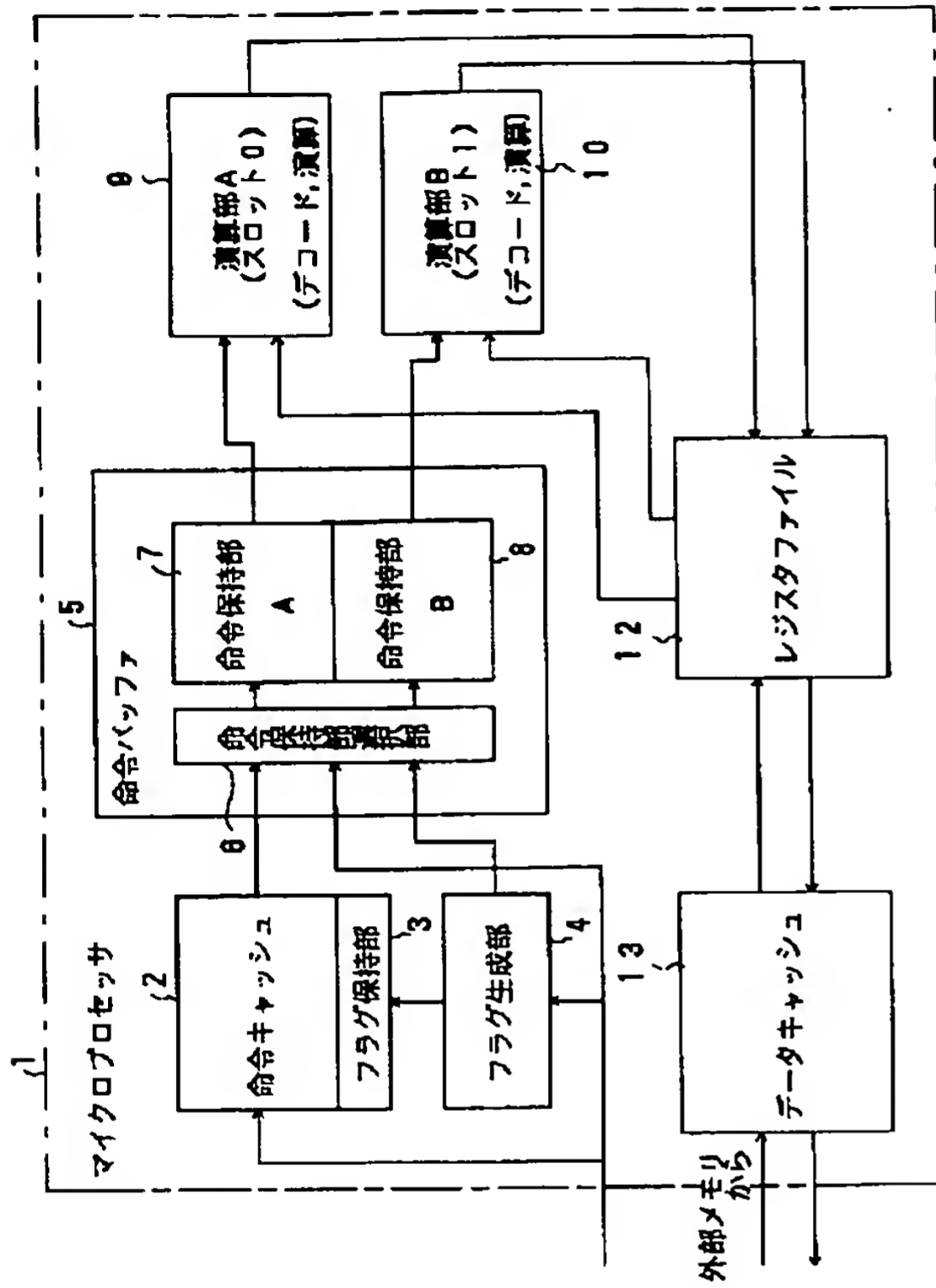
【図5】

命令キャッシュに保持されている命令の例



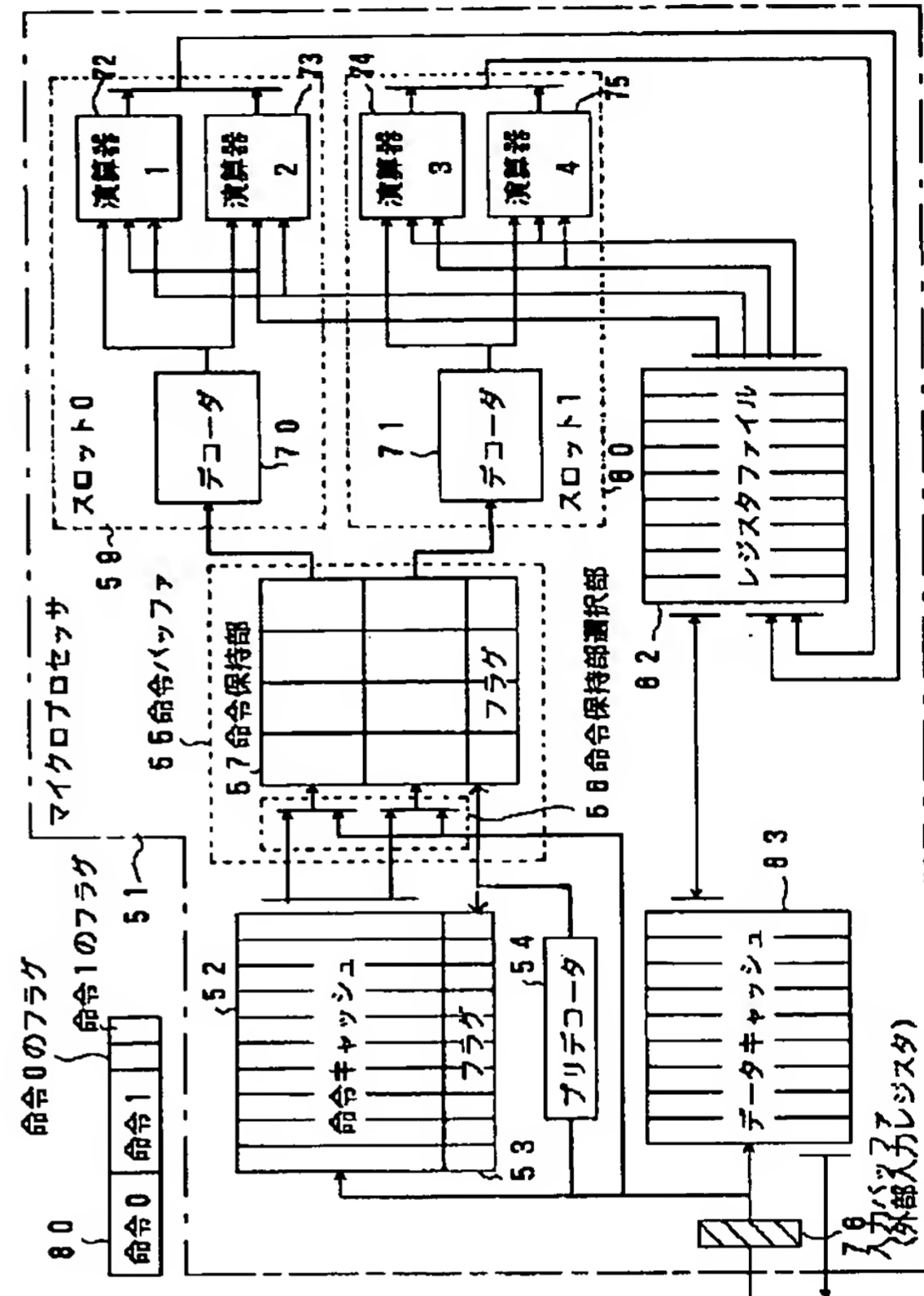
【図 1】

本発明の基本構成



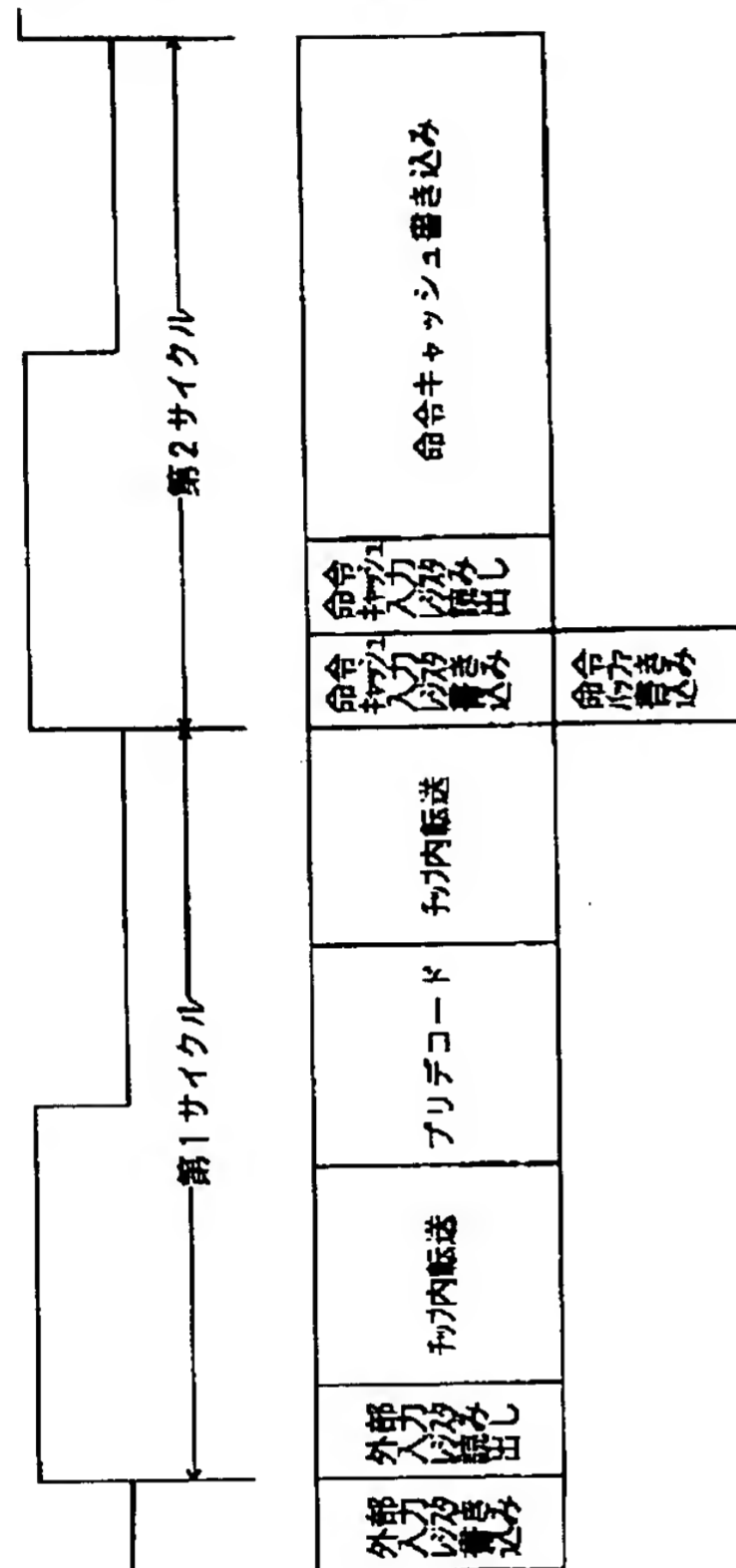
【図 2】

本発明の実施例



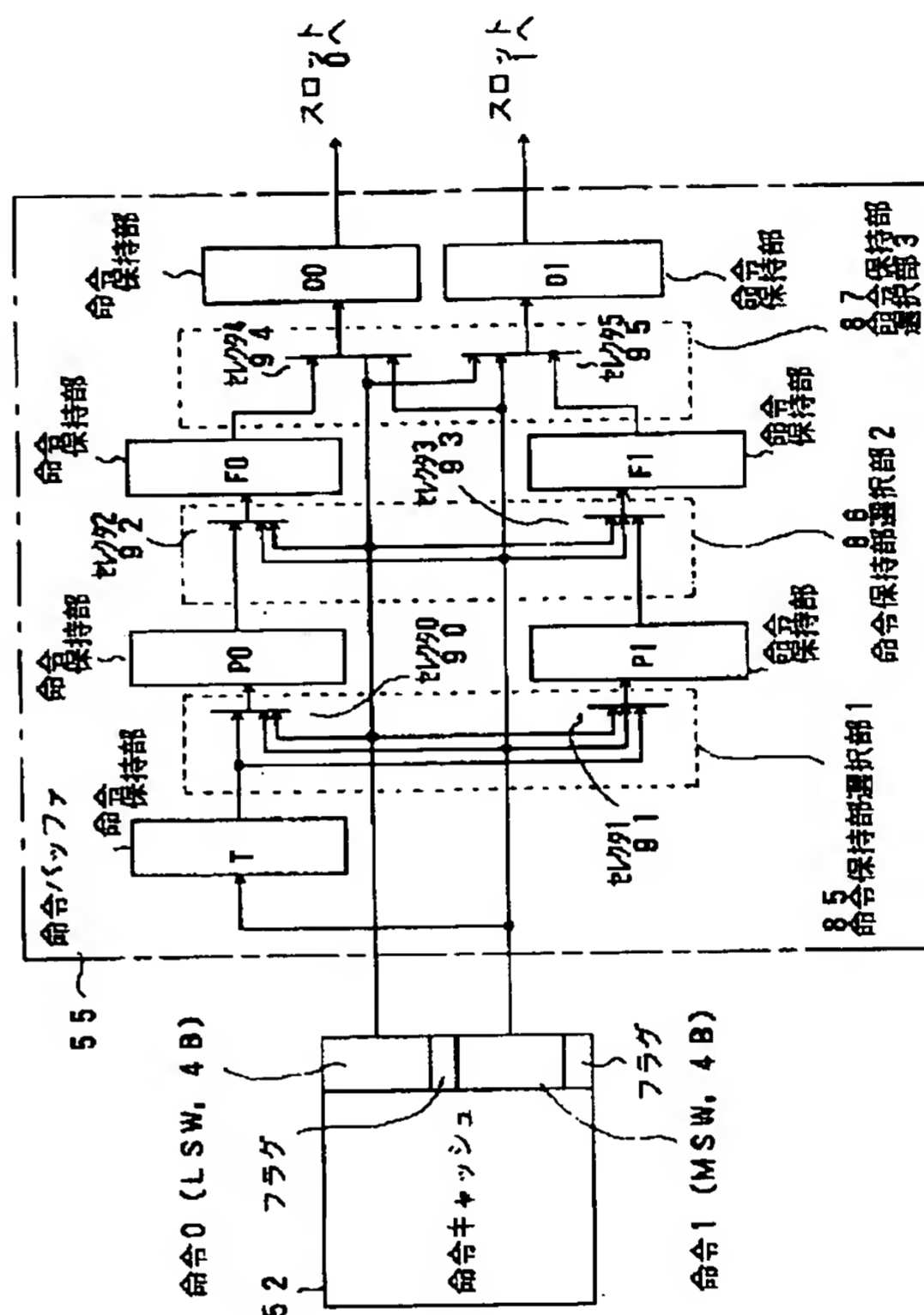
【図 3】

本発明のフラグ生成のタイムチャート



【図 4】

本発明の命令バッファの実施例



【図 6】

動作例 1

| | | | | | | | | | | | |
|---------|-------|----|------|---------|-------|----|------|---------|--|----|------|
| cycle 1 | | | | cycle 2 | | | | cycle 3 | | | |
| | | | T | | | | T | | | | T |
| P1 | | | P0 | P1 | | | P0 | P1 | | | P0 |
| F1 | | | F0 | F1 | | | F0 | F1 | | | F0 |
| D1 | load1 | D0 | add1 | D1 | load2 | D0 | add2 | D1 | | D0 | add3 |
| 2 命令発行 | | | | 2 命令発行 | | | | 1 命令発行 | | | |

| | | | | | | | | | | | |
|---------|-------|----|------|---------|-------|----|------|---------|-------|----|------|
| cycle 4 | | | | cycle 5 | | | | cycle 6 | | | |
| | | | T | | | | T | | | | T |
| P1 | | | P0 | P1 | | | P0 | P1 | | | P0 |
| F1 | load4 | F0 | | F1 | load5 | F0 | add5 | F1 | load6 | F0 | add6 |
| D1 | load3 | D0 | add4 | D1 | load4 | D0 | | D1 | load5 | D0 | add5 |
| 2 命令発行 | | | | 1 命令発行 | | | | 2 命令発行 | | | |

【図 7】

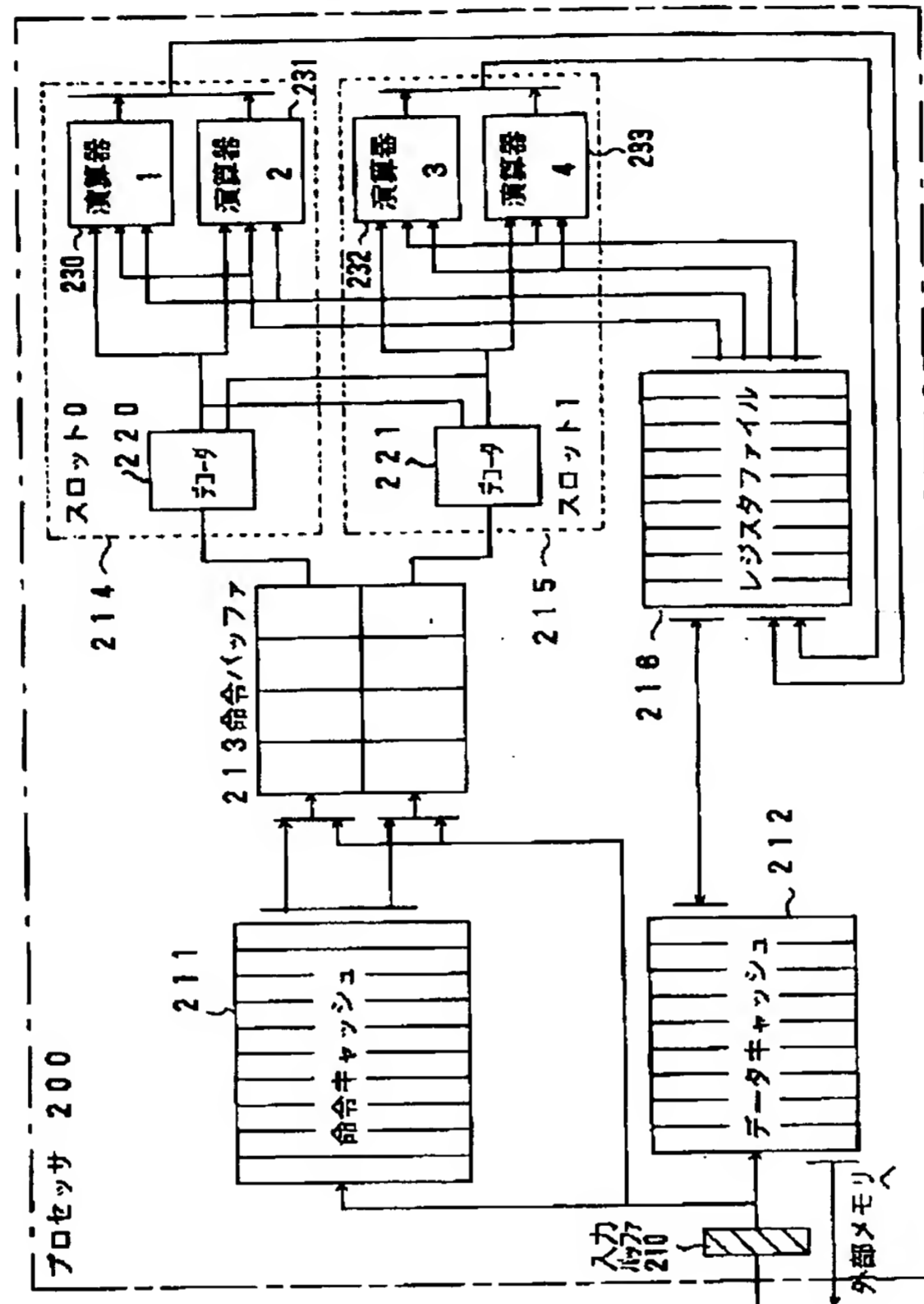
動作例 2

| | | | | | | | | | | | |
|---------|-------|----|------|----------------------------------|-------|----|------|---------|-------|----|------|
| cycle 1 | | | | cycle 2, 3 | | | | cycle 4 | | | |
| | | | T | | | | T | | | | T |
| P1 | | | P0 | P1 | | | P0 | P1 | load3 | P0 | add4 |
| F1 | | | F0 | F1 | load2 | F0 | add2 | F1 | | F0 | add3 |
| D1 | load1 | D0 | add1 | D1 | load1 | D0 | add1 | D1 | load2 | D0 | add2 |
| 0 命令発行 | | | | 0 命令発行 (サイクル2) 2 命令発行 (サイクル3) | | | | 2 命令発行 | | | |

| | | | | | | | | | | | |
|---------|-------|----|------|---------|-------|----|------|---------|-------|----|------|
| cycle 5 | | | | cycle 6 | | | | cycle 7 | | | |
| | | | T | | | | T | | | | T |
| P1 | load4 | P0 | | P1 | load5 | P0 | add5 | P1 | load6 | P0 | add6 |
| F1 | load3 | F0 | add4 | F1 | load4 | F0 | | F1 | load5 | F0 | add5 |
| D1 | | D0 | add3 | D1 | load3 | D0 | add4 | D1 | load4 | D0 | |
| 1 命令発行 | | | | 2 命令発行 | | | | 1 命令発行 | | | |

【図8】

従来のマイクロプロセッサの構成



PATENT ABSTRACTS OF JAPAN

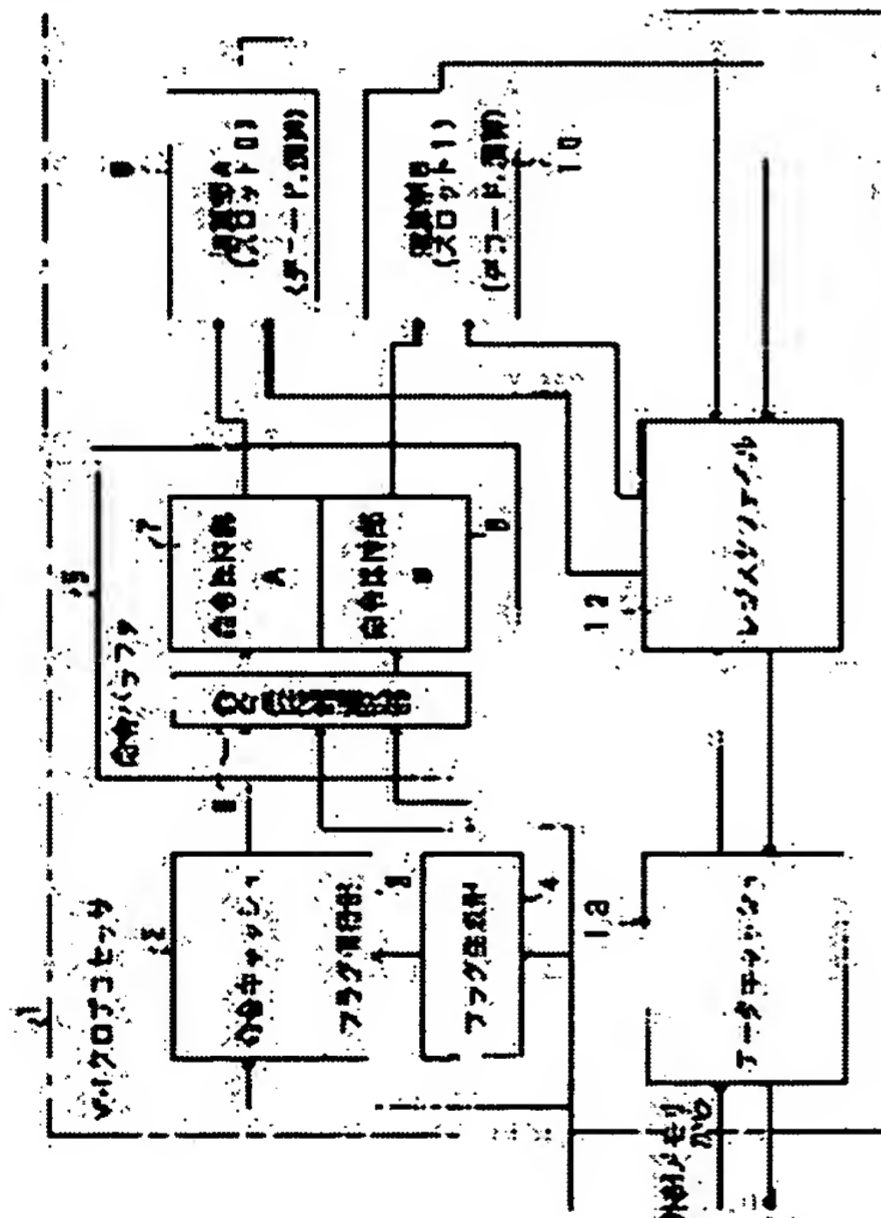
(11)Publication number : 08-044563

(43)Date of publication of application : 16.02.1996

(51)Int.Cl. G06F 9/38
G06F 9/38
G06F 12/08

(21)Application number : 06-178129 (71)Applicant : FUJITSU LTD
(22)Date of filing : 29.07.1994 (72)Inventor : KUBOSAWA HAJIME

(54) MICROPROCESSOR



(57)Abstract:

PURPOSE: To increase the operation speed of a microprocessor by holding an instruction in an instruction holding part corresponding to the operation part, where this instruction is processed, not to select a slot in the stage of decoding.

CONSTITUTION: A flag generation part 4 decodes the instruction inputted from an external memory to discriminate the classification or the instruction. The instruction is read out from an instruction cache 2 into an instruction buffer 5 and is held there. The instruction, which is inputted from the external memory because being absent in the instruction cache 2, is inputted to the instruction buffer 5 after the flag is added by the flag generation part 4, and this instruction is held in the instruction cache 2. In this case, an instruction holding part selection part 6 discriminates the flag; and if the flag is '0', the

instruction buffer 5 is selected, and the instruction is held in an instruction buffer A. If the flag is '1', the instruction holding part selection part 6 selects an instruction holding part B8, and the instruction is held in this part B8.

BEST AVAILABLE COPY